N. 32 FEBBRAIO '88

L. 4.000-Frs. 6,00

# fare ELETTRONICA

Realizzazioni pratiche • TV Service • Radiantistica • Computer hardware

## REALIZZAZIONI PRATICHE

**Telecomando via rete**

**Generatore di funzioni**

## COMPUTER HARDWARE

**Apriporta codificato**

## RADIOCOMANDO PROPORZIONALE A 5 CANALI

## RADIANTISTICA

**Semplice generatore sweep**

## TV SERVICE

**Zanussi BS290.1**

# UN APRIPORTA CODIFICATO

di A. Ginghiali, e M. Ferdeghini del C.N.R. di Pisa

In tutti quei casi in cui si renda necessario consentire l'accesso a determinati locali, soggetti a particolari condizioni di sicurezza, solo ad un limitato numero di persone è ormai consuetudine utilizzare sistemi di apertura di porte utilizzanti delle "chiavi elettroniche" sia di tipo hardware (telecomandi) sia di tipo software (lettura di codici con sistemi magnetici od ottici). Il nostro sistema è una variante di quest'ultimo genere perchè richiede l'uso di un codice da battere su un tastierino, che fa capo a un microcontrollore. L'uso del microcomputer e di una stampantina offre anche la possibilità di mettere a disposizione, per eventuali controlli, una documentazione scritta sul personale che ha inteso usufruire dell'accesso ai suddetti locali. A nostro giudizio questo sistema rappresenta un'alternativa poco costosa a sistemi più sofisticati già citati, mantenendo comunque equivalenti caratteristiche di affidabilità. Vediamo rapidamente qual' è il protocollo da seguire per abilitare l'azionamento del servomeccanismo di apertura di porta d'accesso, a cui il nostro sistema è stato applicato. Attraverso una tastiera viene impostato dall'utente il proprio codice a cinque cifre; in caso di errore, premendo uno qualsiasi dei tasti di correzione d'immissione, si può ricominciare a reimpostare il codice. Un microcomputer effettua quindi una ricerca nel set di codici autorizzati; in caso venga verificata la correttezza del codice immesso, viene abilitata l'apertura della porta; contemporaneamente il nome della persona corrispondente al codice immesso, l'ora e la data del suo ingresso sono stampati su di un supporto cartaceo. Analizziamo ora la parte hardware costituente il sistema. Per quanto riguarda la stampante la scelta è caduta sul modello DP 505 della CITIZEN che dispone di un data bus di tipo parallelo Centronics ed ha un formato di 24 colonne con varie possibilità di scrittura (due colori, caratteri allargati e rovesciati). La stampante e la sua scheda controllo necessitano per la loro alimentazione di una tensione di 12 Vcc con corrente max di 3 A per l'alimentazione del motore della stampante, e di una tensione di 5 Vcc per l'alimentazione della scheda. Nel nostro caso queste tensioni sono state ambedue ricavate da un alimentatore da 12 V e 6 A con il quale viene alimentato direttamente il motore della stampante, mentre la tensione di 5 Vcc necessaria per la scheda controllo viene ricavata mediante l'impiego di un integrato stabilizzatore di tipo 7805; da questo alimentatore viene anche prelevata anche la tensione di alimentazione della scheda controllo a microprocessore. La tastiera è del tipo a film a matrice della Mecanorma a 12 tasti di

```
PROGRAMMA DOORLOCK

        *=$0
PA      *=*+1
PB      *=*+1
PC      *=*+1
PD      *=*+$F
IER     *=*+2
MCR     *=*+4
LTA     *=*+2
UTA     *=*+2
        *=$65
PCODL   *=*+2
TASTO   *=*+1
CODE    *=*+5
        *=$40
TIMO    *=*+1
ANNO    *=*+2
MESE    *=*+1
GIORNO  *=*+1
MIN     *=*+1
HOR     *=*+1
SECN    *=*+1
;
        *=$9000
; INIZIALIZZAZIONI
RESET   SEI
        CLD
        LDX #$FF
        TXS
        STX PA
        STX PB
        LDX #%10111110
        STX PC
; 4 PORTE, ADDRESS BUS RIDOTTO,
; PORT D OUTPUT
        LDA #$60
        STA MCR
; TIMER 1-INTERRUPT ATTIVO
        LDA #$10
        STA IER
        LDA #<49998
        STA LTA
; MESSAGGIO INIZIALE
        LDX #0
MESS    LDA STMSG,X
        PHA
        AND #$7F
        STA PD
        JSR BUSY
        INX
        PLA
        BPL MESS
        LDA #>50000
        STA UTA
INIT    CLI
; LOOP ATTESA KEYBOARD
        SMB0 PC
        LDX #0
LOOP3   TXA
        PHA
        JSR TAST
        PLA
        TAX
        RMB0 PC
        LDA TASTO
; TASTO DI ANNULLO INPUT OPPURE
; TASTO NON PREMUTO
        CMP #$FF
        BEQ INIT
; MEMORIZZA CODICE
        STA CODE,X
        INX
        CPX #5
        BNE LOOP3
; INPUT CODICE COMPLETATO
; RICERCA IN LIBRERIA CODICI
        LDA #<LISTA
        STA PCODL
        LDA #>LISTA
        STA PCODL+1
LOOP4   LDY #0
LOOP5   LDA (PCODL),Y
; FINE LISTA?
        CMP #$FF
        BEQ INIT
        CMP CODE,Y
; CONTINUA CODICE
        BEQ NEXT
; CERCA NUOVO CODICE
        CLC
        LDA PCODL
        ADC #$19
        STA PCODL
        LDA PCODL+1
        ADC #0
        STA PCODL+1
        JMP LOOP4
NEXT    INY
        CPY #5
        BNE LOOP5
; CODICE TROVATO
; APRE PORTA
RICOD   SMB6 PC
; STAMPA NOME
LOOP6   LDA (PCODL),Y
        PHA
        AND #$7F
        STA PD
        JSR BUSY
        INY
        PLA
        BPL LOOP6
; STAMPA ORA & DATA
        JSR STDAT
; DOPPIO LINEFEED
        LDA #$C
        STA PD
        JSR BUSY
        LDA #2
        STA PD
        JSR BUSY
; FINE COMANDO APERTURA PORTA
        RMB6 PC
; RICOMINCIA
        JMP INIT
; ROUTINE INVIO DATI A PRINTER
BUSY    BBS0 PA,BUSY
        RMB7 PC
        SMB7 PC
        RTS
; ROUTINE DI STAMPA ORA E DATA
STDAT   LDX #6
STD1    LDA TIMO,X
        PHA
        AND #$F0
        LSR A
        LSR A
        LSR A
        LSR A
        ORA #$30
        STA PD
        JSR BUSY
        PLA
        AND #$7F
        ORA #$30
        STA PD
        JSR BUSY
        DEX
        BEQ ESC
        CPX #5
        BNE STD2
        LDA #':
        STA PD
        JSR BUSY
        BNE STD1
STD2    CPX #4
        BNE STD3
        LDA #' '
        STA PD
        JSR BUSY
        BNE STD1
STD3    CPX #1
        BEQ STD1
        LDA #'/
        STA PD
        JSR BUSY
        BNE STD1
ESC     LDA #$D
        STA PD
        JSR BUSY
        RTS
; ROUTINE LETTURA KEYBOARD
TAST    LDA #$F
        STA PB
LOOPB   LDA PB
        AND #$F
        CMP #$F
        BEQ LOOPB
        LDA PB
        PHA
        LDA #$70
        STA PB
        LDA PB
        AND #$70
        LDX #0
        CMP #$60
        BEQ LIN
        LDX #1
        CMP #$50
        BEQ LIN
        LDX #2
LIN     PLA
        AND #$F
        LDY #0
        CMP #$E
```

**Figura 1. Programma in assembler per il controllo della scheda dell' apriporta**

cui 10 vengono usati come tasti numerici e 2 come reset di immissione nel caso di errori di battitura; questa tastiera è stata scelta sia per il suo costo piuttosto contenuto che per la praticità che offre in quanto essendo autoadesiva è possibile applicarla in maniera estremamente semplice dovunque si renda necessario. La scheda controllo a microprocessore, della quale analizzeremo tra breve il funzionamento, è stata realizzata basandosi sul microprocessore 6501AQ della Rockwell che dispone di un set di istruzioni praticamente uguale a quello del più noto 6502 con

**Figura 1. Seguito del programma in assembler**

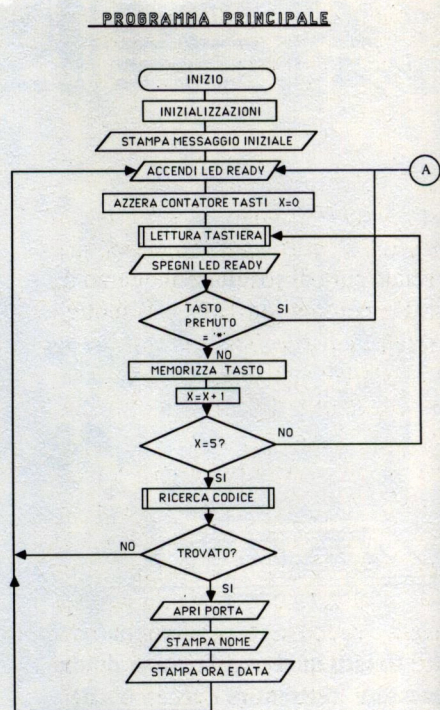
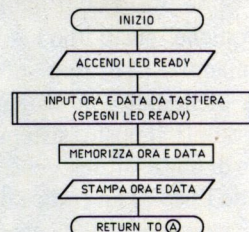
```
        BEQ NUM
        LDY #1
        CMP #$D
        BEQ NUM
        LDY #2
        CMP #$B
        BEQ NUM
        LDY #3
NUM     LDA #0
        STA TASTO
        CLC
        INY
LOOPY   DEY
        BEQ XXX
        LDA TASTO
        ADC #3
        STA TASTO
        JMP LOOPY
XXX     TXA
        ADC TASTO
        STA TASTO
LOOPW   LDA #$F
        STA PB
        LDA PB
        CMP #$F
        BNE LOOPW
        LDA TASTO
        TAY
        LDA PTAB,Y
        STA TASTO
        RTS
; TABELLA CONVERSIONE KEYBOARD
PTAB    BYT $FF,'0',$FF,'123456789'
; AGGIORNAMENTO ORA E DATA
IRQ     PHA
        TXA
        PHA
        TYA
        PHA
        SED
        CLC
        LDA TIMO
        ADC #1
        STA TIMO
        CMP #$20
        BCS IR1
        IMP RETUR
IR1     LDA #0
        STA TIMO
        LDA SECN
        ADC #0
        STA SECN
        CMP #$60
        BCC RETUR
        LDA #0
        STA SECN
        LDA MIN
        ADC #0
        STA MIN
        CMP #$60
        BCC RETUR
        LDA #0
        STA MIN
        LDA HOR
        ADC #0
        STA HOR
        CMP #$24
        BCC RETUR
        LDA #0
        STA HOR
        LDA GIORNO
        ADC #0
        STA GIORNO
        TAX
        LDA MESE
        CMP #4
        BEQ TRENT
        CMP #6
        BEQ TRENT
        CMP #9
        BEQ TRENT
        CMP #11
        BNE FEBBR
TRENT   CPY #$31
        BCC RETUR
        BCS INMS
FEBBR   CMP #2
        BNE TREN1
        BEQ BIS
        LSR A
        BCS NOBIS
        LSR A
        BCC BIS
NOBIS   CPX #$29
        BCS INMS
        BCC RETUR
BIS     CPX #$30
        BCS INMS
        BCC RETUR
TREN1   CPX #$32
        BCC RETUR
INMS    LDA #1
        STA GIORNO
        LDA MESE
        ADC #0
        STA MESE
        CMP #$13
        BCC RETUR
        LDA #1
        STA MESE
        LDA ANNO
        ADC #0
        STA ANNO
        BCC RETUR
        LDA ANNO+1
        ADC #0
        STA ANNO+1
; RESET TIMER
RETUR   LDA LTA
        PLA
        TAY
        PLA
        TAX
        PLA
        RTI
; INTRODUZIONE ORA E DATA
DATIN   SMB0 PC
        JSR LOOP1
        CMP #$24
        BCS RT1
        STA HOR
        JSR LOOP1
        CMP #$60
        BCS RT1
        STA MIN
        JSR LOOP1
        CMP #$32
        BCS RT1
        STA GIORNO
        JSR LOOP1
        CMP #$13
        BCS RT1
        STA MESE
        JSR LOOP1
        CMP #$21
        BCS RT1
        CMP #$19
        BCC RT1
        STA ANNO+1
        JSR LOOP1
        STA ANNO
RT1     JSR STDAT
        LDA #$C
        STA PD
        JSR BUSY
        LDA #2
        STA PD
        JSR BUSY
        PLP
        PLA
        PLA
        LDA #>INIT
        PHA
        LDA #<INIT
        PHA
        PHP
        RTI
LOOP1   JSR TAST
        RMB0 PC
        LDA TASTO
        CMP #$FF
        BEQ LOOP1
        AND #$F
        ROL A
        ROL A
        ROL A
        ROL A
        STA CODE
        JSR TAST
        LDA TASTO
        CMP #$FF
        BEQ LOOP1
        AND #$F
        ORA CODE
        RTS
STMSG   .BYT $11
        .BYT '======================,$D
        .BYT 'INTERRUZIONE        ',$D
        .BYT 'ALIMENTAZIONE       ',$D
        .BYT $13,$E,'AGGIORNARE '
        .BYT $F,$D,$13,$E,'ORARIO    '
        .BYT $F,$D
        .BYT '======================',$8D
LISTA   .BYT '17171   ','GINGHIALI',$8D
        *=*+8
        .BYT '34567   ','FERDEGHINI',$8D
        *=*+7
        .BYT '12670    ','ALTRINOMI',$8D
        *=$9FFA
        .WOR DATIN,RESET,IRQ
        .END
```

ROUTINE NMI PER INTRODUZIONE DATA E ORA

ROUTINE IRQ PER AGGIORNAMENTO ORA E DATA

**Figura 2. Diagramma di flusso del programma di controllo e delle routine di inserimento e aggiornamento dell' ora.**

l'aggiunta di alcune istruzioni di manipolazione di singoli bit che sono state usate nel programma (si veda il listing in Figura 1). NOTA: è importante per la realizzazione di questa scheda utilizzare il microprocessore 6501AQ in quanto il modello equivalente 6501Q opera una diversa divisione sulla frequenza di clock generata dall'oscillatore interno, tut-

Figura 3. **Schema elettrico della scheda dell' apriporta, la sua programmazione si esegue per mezzo di un tastierino.**

tavia nel caso in cui si abbia a disposizione il secondo modello sarà possibile utilizzarlo ugualmente avendo cura di sostituire il quarzo da 2MHz con uno da 4MHz. Il motivo principale per cui è stato scelto questo microprocessore consiste nella presenza di 4 porte di I/O e di una piccola, ma sufficiente per i nostri scopi, memoria RAM sul chip del microprocessore stesso, cosicchè si rende possibile una notevole semplificazione della circuitistica evitando circuiti di interfaccia, memorie RAM e relativi decoder. Il programma ed i dati sono stati memorizzati su di una memoria EPROM di tipo 2532 della

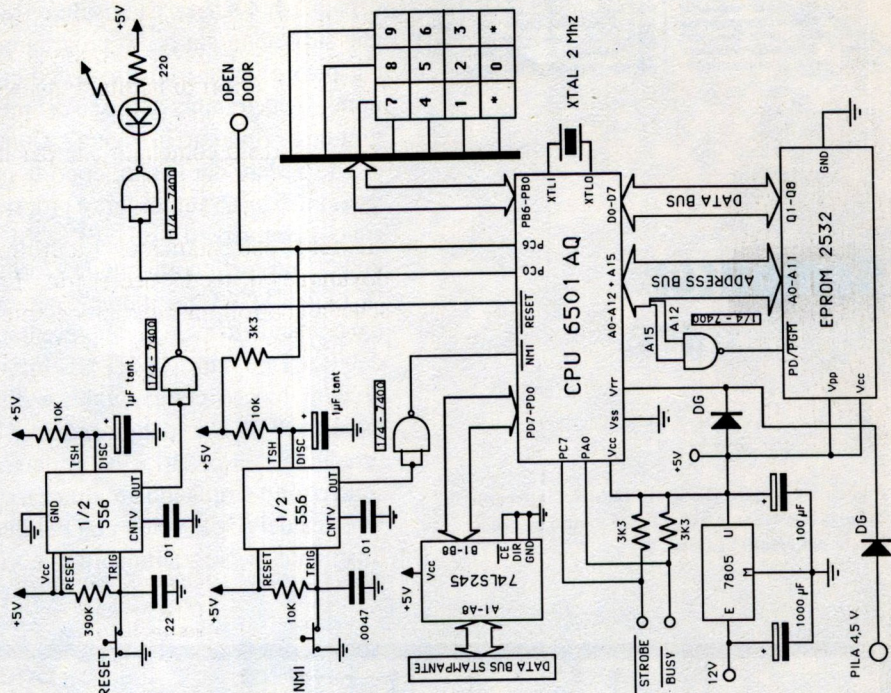
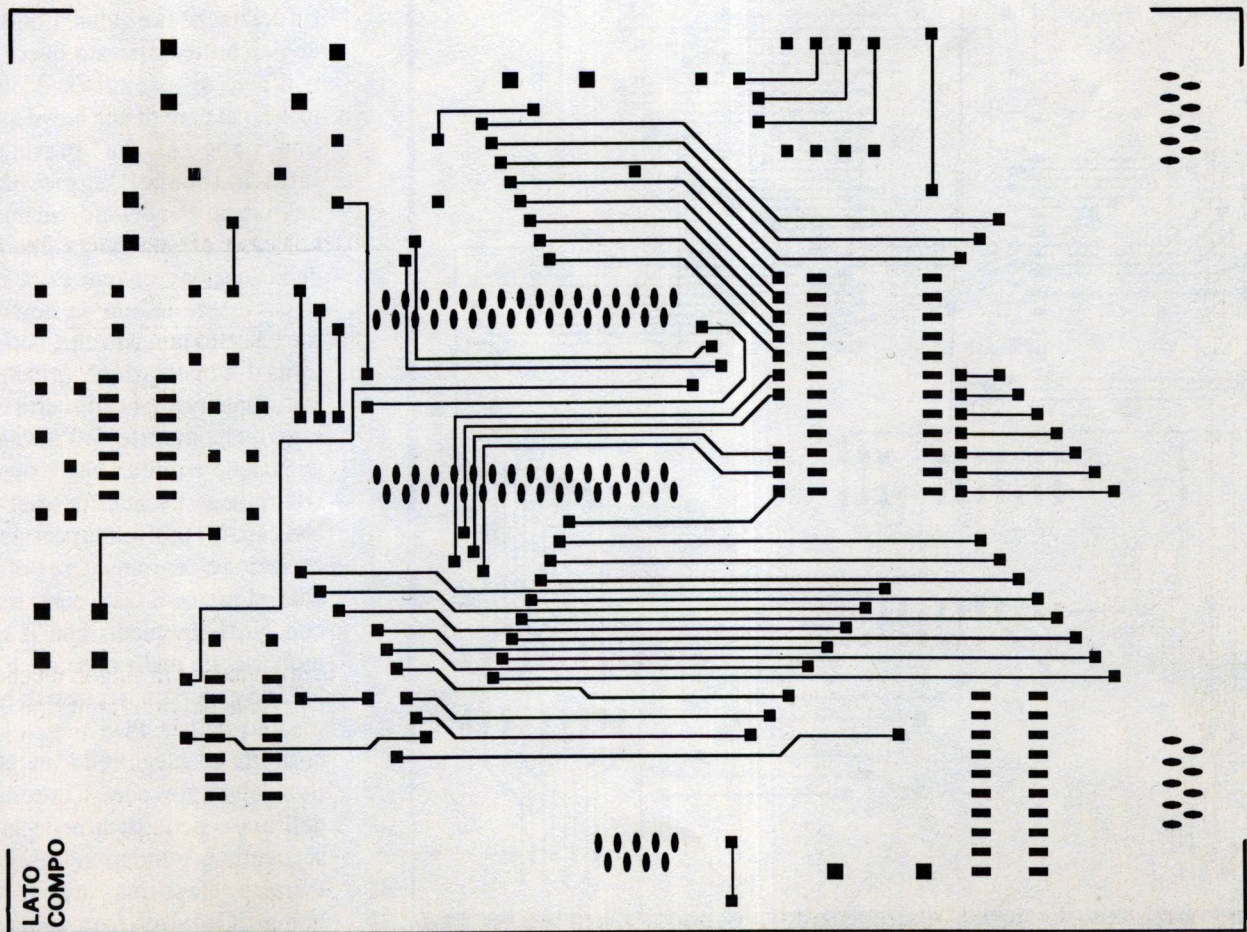
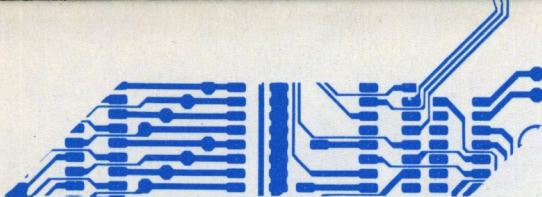



**Figura 4. Piste ramate del circuito stampato in scala naturale presenti sul lato componenti.**

LATO COMPO

**Figura 5. Circuito stampato visto dal lato rame in scala naturale. Le due facce vanno intercollegate nei relativi punti.**

LATO RAME

capacità di 4 Kbyte, che risultano più che sufficienti sia per il programma che per un elevato numero di codici e nomi. Il diagramma di flusso del programma è riportato in Figura2. Come si può vedere dal listato, dopo le istruzioni di programma vere e proprie, sono memorizzati sulla EPROM i codici ed i nomi nella forma "C1,C2,C3,C4,C5,-,-,nome", seguiti dal comando di stampa rappresentato dal "carriage return" ( $D ). L'inizio di ogni blocco codice-nome è distanziato di 25 byte dal precedente. Il problema dell'orario e della data è stato risolto sfruttando un'altra caratteristica del 6501 e cioè uno dei due timer interni programmabili che viene utilizzato in modo da generare una richiesta di interrupt (IRQ) ogni 50ms; la corrispondente routine provvede ad aggiornare il valore dei minuti, ore, giorni, mesi, anni tenendo conto anche degli anni bisestili fino all' anno 2100. Una peculiarità del 6501, cioè quella di avere una parte della RAM interna alimentata separatamente dall'alimentazione principale permettendo così il collegamento di una batteria tampone, ci ha consentito di limitare i problemi derivanti da eventuali interruzioni dell'alimentazione. Come si può vedere dallo schema elettrico riportato in Figura 3, il suddetto collegamento è stato effettuato mediante i due diodi al germanio DG i quali collegano praticamente a Vcc ( a parte la tensione di soglia del diodo) il piedino Vrr di alimentazione delle memorie, quando l'alimentazione principale è presente, e collegano Vrr alla batteria tampone da 4,5 V, quando l'alimentazione principale viene a mancare. Infatti per evitare che una qualunque interruzione della tensione di rete anche della durata di pochi secondi potesse cancellare la memoria della scheda e quindi l'ora e la data abbiamo memorizzato quest' ultime nell'area di memoria suddetta, cosicché, nel caso di una breve interruzione, non si ha praticamente interruzionenell'aggiornamento dell'orario. Per evitare comunque che nel caso di una lunga interruzione della tensione di rete l'ora indicata possa essere affetta da notevole errore, è stato previsto che, non appena torna l'alimentazione principale, la stampante stampi sulla carta un messaggio che avverte dell'avvenuta interruzione e quindi che è opportuno controllare l'esattezza dell'orario. Nel caso in cui si renda necessaria una correzione, per immettere l'orario e la data si preme il tasto contrassegnato con NMI, in modo che il circuito monostabile realizzato con 1/2 di un NE556 generi un impulso di NMI sul piedino della CPU; questa è allora costretta ad eseguire la routine relativa che prevede l'introduzione dell'ora e della data nelle modalità seguenti. L'immissione è effettuata tramite tastiera nella forma hhmmGGMMAAAA; appena con-

clusa l'operazione, il che avviene con l'introduzione dell'ultimo carattere, vengono stampate sulla carta l'ora e la data in modo da poterne controllare la correttezza; nel caso di errori si preme di nuovo il tasto NMI e si ripete la procedura. Se l'errore è commesso a metà della battitura di una coppia di dati, si può correggere premendo uno dei tasti di reset d'immissione, e procedere di seguito, senza ripartire da capo. Analizziamo ora più in dettaglio lo schema elettrico della scheda stessa riportato in Figura 3. In alto vediamo i due circuiti monostabili, utilizzati per gli impulsi di NMI e RESET, che sono stati relizzati come già detto con un solo integrato di tipo 556 il quale come noto contiene al suo interno 2 temporizzatori di tipo 555; i due condensatori da 1µF sono stati scelti del tipo al tantalio per limitare l'ingombro sulla scheda stessa ma comunque possono essere utilizzati anche altri tipi purché di capacita equivalente. Il LED che vediamo sulla destra serve come segnalatore di "READY" e cioè indica che il circuito è pronto a ricevere una battitura di codice, nel caso in cui tale LED risulti

**Figura 6. Disposizione dei componenti sul circuito stampato.**

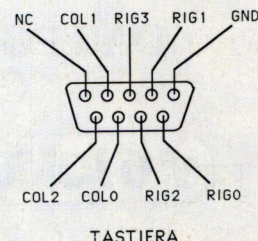
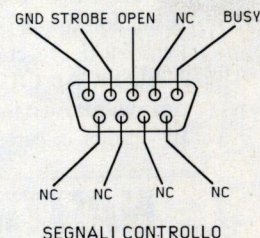
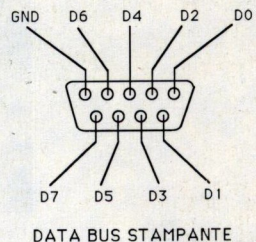
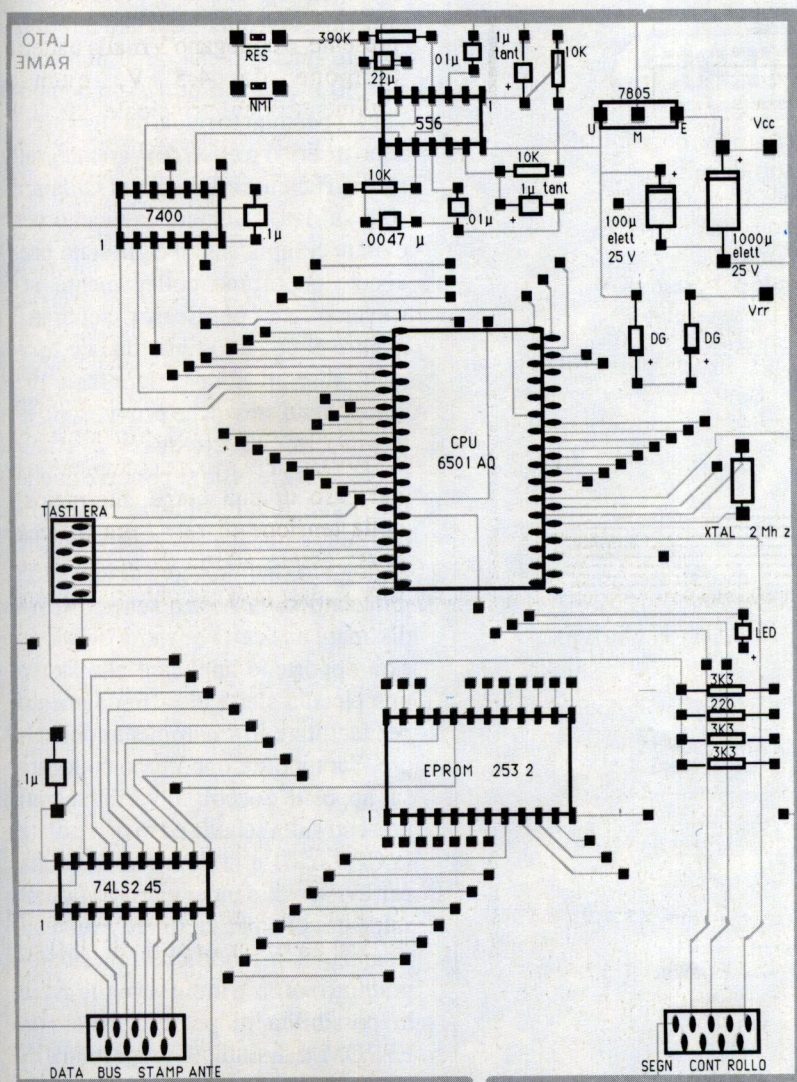



**Figura 7. Piedinatura dei vari connettori visti di fronte**

spento si dovrà, prima di battere il proprio codice sulla tastiera, premere sulla tastiera stessa uno qualsiasi dei due tasti di reset d'immisione. Più in basso sulla sinistra c'è da notare la presenza di un 74LS245 che svolge la funzione di buffer per i dati in uscita verso il data bus della stampante, e dei due terminali contrassegnati con "STROBE" e "BUSY" che andranno collegati alle analoghe linee sul control bus della stampante. L'uscita contrassegnata con OPEN DOOR, che vediamo sulla destra, comanda con un impulso positivo il servomeccanismo che apre la porta di accesso nel caso in cui il codice sia stato riconosciuto. Per quanto riguarda l'alimentazione della scheda a microprocessore, questa viene ottenuta, come già spiegato, dall'alimentatore a 12V mediante l'impiego di un integrato stabilizzatore di tipo 7805. I due diodi DG, attraverso i quali giunge

tensione al piedino di alimentazione dell'area di memoria prima menzionata, debbono essere di tipo al germanio in quanto i tipi al silicio hanno una tensione di soglia troppo elevata che potrebbe far giungere al piedino Vrr dell CPU una tensione insufficiente. Passiamo ora ad alcune considerazioni di carattere patico riguardanti la realizzazione pratica della scheda. Il circuito stampato che risulta del tipo a doppia faccia e che possiamo vedere nelle figura 4, 5 è stato concepito in modo da non rendere necessaria la metallizzazione dei fori, ciò sia per consentire una maggiore economia che per difficoltà di carattere pratico riguardanti la realizzazione di uno stampato di tale tipo. Come prima operazione si dovrà quindi provvedere a collegare con uno spezzone di filo le piste del lato superiore con quelle del lato inferiore in tutti quei fori nei quali non sono inseriti i terminali di nessun componente; per quanto riguarda i fori in cui sono inseriti i piedini dei vari integrati sarà sufficiente effettuare la saldatura solo sul lato saldature in quanto per evitare complicazioni è già stato previsto l'opportuno collegamento attraverso un ponticello separato, mentre per quanto riguarda i componenti discreti sono i terminali dei componenti stessi che provvedono in alcuni casi ad effettuare il collegamento trà le piste superiori e quelle inferiori e quindi debbono essere saldati da ambedue le parti. Poiché l'integrato stabilizzatore di tipo 7805 utilizzato sulla nostra scheda dovrà dissipare una certa quantità di calore, sarà opportuno applicare allo stesso una piccola aletta di raffreddamento per facilitare lo smaltimento del calore. Sarà inoltre opportuno collocare su appositi zoccoli i vari integrati presenti sulla scheda ed in particolare la CPU 6501 e la EPROM 2532, sia per evitare di danneggiarli in fase di saldatura che per facilitare eventuali riparazioni o modifiche (questo soprattutto nel caso in cui si voglia avere la possibilità di poter cambiare la EPROM e quindi il programma di controllo per eventuali usi diversi da quello originale o semplicemente per aggiungere o togliere nominativi al set di codici e persone autorizzate).

# IL C64 COME FREQUENZIMETRO

**di P. Loddo**

La porta d'utente del C64, facile da programmare, permette di utilizzare agevolmente questo computer come strumento di misura. Dopo aver visto nei numeri scorsi il C64 come strumento di misura e come tester di IC TTL, potrete ora usare il vostro fedele computer per mettere assieme, con estrema facilità, un frequenzimetro digitale, per il quale sono necessari soltanto pochi componenti hardware.

Per lavorare in hardware è necessario disporre di alcuni strumenti, tra i quali il frequenzimetro digitale ha un ruolo importante, perchè permette di misurare con esattezza la frequenza di un segnale. Un simile strumento è molto utile per tarare esattamente la frequenza degli oscillatori, oppure quando sia necessario contare con precisione il numero degli impulsi che si susseguono in un determinato intervallo di tempo (per esempio quelli emessi da una barriera fotoelettrica). In questo articolo troverete una semplice e potente soluzione per misurare le frequenze con il C64, anche nella versione base priva di estensioni hardware.

Per il conteggio delle frequenze ai livelli TTL (0 V e +5 V), esiste per il C64 una soluzione esclusivamente software. Il programma necessario si compone di due parti: una in BASIC ed una in linguaggio macchina. Iniziamo con alcuni chiarimenti riguardanti la parte in BASIC.

Quando avrete impostato il programma BASIC riportato nel listato 1, trasferitelo immediatamente in una memoria permanente, perchè dovrà essere caricato dopo l'avviamento del programma in linguaggio macchina presentato nel listato 2, che in un primo momento non è ancora disponibile. Nel programma BASIC viene stabilito, tramite menù, l'intervallo di conteggio, cioè il tempo utilizzato dal computer per misurare la frequenza. La scelta sarà possibile tra 1/128 e 128 s. Una volta trascorso l'intervallo di conteggio, la frequenza in hertz viene visualizzata nella parte alta dello schermo ed il relativo valore viene aggiornato ad ogni intervallo di conteggio.

**Figura 1. Circuito per misurare piccoli segnali.**

Se la frequenza è rapidamente variabile, viene visualizzato il valore medio nell'ambito di un intervallo di conteggio. Premendo il tasto SHIFT, questo processo di misura e visualizzazione viene interrotto e potrete così scegliere un diverso intervallo di conteggio. Poichè il tasto premuto viene comunque analizzato dopo la fine dell'intervallo di conteggio, nel caso di intervalli lunghi è consigliabile premere il tasto SHIFT/LOCK, attenzione però: questo tasto non deve essere più premuto nella successiva scelta sul menù. Con RUN/STOP + RESTORE potrete interrompere il programma anche durante il processo di misura. Il programma BASIC inserisce negli indirizzi da 704 a 708 (da $02C0 a $02C4) i parametri per il programma in linguaggio macchina, che verranno in seguito prelevati da queste

posizioni. La parte in linguaggio macchina del listato 2 ha il nome di file "52000count.obj" ed è inserita nella memoria tra gli indirizzi 56000 e 52236 (da $CB20 a $CC0C). Nel programma, vengono utilizzati per la sincronizzazione i temporizzatori di CIA1, mentre il temporizzatore A di CIA2 serve al conteggio degli impulsi. Il programma utilizza allo scopo le interruzioni ed impedisce l'analisi della tastiera durante l'intervallo di conteggio. Per sveltire al massimo il procedimento di misura, l'interruzione non viene introdotta in $0314 tramite il vettore IRQ, ma è prelevata direttamente dalla locazione RAM $0314. Poichè il programma in linguaggio macchina produce poi nuovamente la normale routine di interruzione, non sono prevedibili problemi di compatibilità con i floppy speeder.

L'ingresso per la tensione alternata da misurare è costituito dal piedino 6 della porta di utente, mentre la massa potrà essere prelevata dal piedino 1. Si possono applicare direttamente i segnali TTL alla porta di utente e lo stesso potrà avvenire con tensioni alternate di uguale livello (eventualmente ottenuto utilizzando un partitore di tensione).

Le tensioni alternate di minore livello, come i segnali audio provenienti da un registratore a cassette, devono essere dapprima condizionati mediante un amplificatore ed un trigger di Schmitt. Lo stesso risultato può essere raggiunto con il circuito di Figura 1. Un invertitore logico viene usato come amplificatore, secondo un sistema alquanto inconsueto. Analogamente agli amplificatori operazionali, il resistore di retroazione determina la sensibilità del circuito, un suo valore resistivo maggiore aumenta la sensibilità.

Lo stadio successivo, con due coppie invertitore-resistore di retroazione, provvede a garantire i livelli TTL. Il successivo invertitore serve ad eliminare l'inversione causata dal numero dispari dei precedenti stadi.

Il circuito può essere realizzato con un integrato 4069, che contiene sei invertitori e si può facilmente montare su una basetta preforata per prototipi. La piedinatura di questo circuito integrato è disegnata in Figura 2.

Gli ingressi degli invertitori non utilizzati (piedini 11 e 13) non devono essere lasciati aperti (pericolo di oscillazioni parassite), ma dovranno essere collegati alla linea dei +5 V, in modo che le uscite vadano a massa. Le frequenze acustiche da misurare potranno essere prelevate, per esempio, da un televisore durante la ricezione del monoscopio. Con il programma di misura potanno essere misurate frequenze fino a 1/4 del clock di sistema, cioè più di 200 kHz. C'è un particolare che merita la massima attenzione: non dovete applicare tensioni ignote (eccessive o negative) direttamente alla porta di utente, perchè potrebbero danneggiare il CIA.

**Figura 2. Piedinatura del circuito integrato CMOS 4069 (vista dall'alto)**

### ELENCO DEI COMPONENTI

| | |
|---|---|
| **R1-3** | **resistori da 10 kΩ 1/4 W 5%** |
| **R2-4** | **resistori da 100 kΩ 1/4 W 5%** |
| **C1** | **condensatore da 10 nF ceramico a disco** |
| **C2** | **condensatore da 1 μF al tantalio** |
| **IC1** | **circuito integrato 4069** |

### Tabella di conversione

```
{HOME}.........HOME
{CLR}..........PULIZIA SCHERMO
{CUR.SU}.......CURSORE IN ALTO
{CUR.GIU}......CURSORE IN BASSO
{CUR.DES}......CURSORE A DESTRA
{CUR.SIN}......CURSORE A SINISTRA
{SPC}..........SPAZIO
{RVS ON}.......REVERSE ON
{RVS OFF}......REVERSE OFF
{INST}.........INSERT
{F1}...........TASTO F1
{F2}...........TASTO F2
{F3}...........TASTO F3
{F4}...........TASTO F4
{F5}...........TASTO F5
{F6}...........TASTO F6
{F7}...........TASTO F7
{F8}...........TASTO F8
{BLACK}........COL. NERO (CTRL+1)
{WHITE}........COL. BIANCO (CTRL+2)
{RED}..........COL. ROSSO (CTRL+3)
{CYAN}.........COL. CIANO (CTRL+4)
{PURPLE}.......COL. PORPORA (CTRL+5)
{GREEN}........COL. VERDE (CTRL+6)
{BLUE}.........COL. BLU (CTRL+7)
{YELLOW}.......COL. GIALLO (CTRL+8)
{ORANGE}.......COL. ARANCIO (CBM+1)
{BROWN}........COL. MARRONE (CBM+2)
{LT.RED}.......COL. ROSSO CHIARO (CBM+3)
{GRAY1}........COL. GRIGIO 1 (CBM+4)
{GRAY2}........COL. GRIGIO 2 (CBM+5)
{LT.GREEN}.....COL. VERDE CHIARO (CBM+6)
{LT.BLUE}......COL. BLU CHIARO (CBM+7)
{GRAY3}........COL. GRIGIO 3 (CBM+8)

 I CARATTERI GRAFICI, OTTENUTI CON LA
PRESSIONE DEI TASTI 'SHIFT' E 'CBM',
SONO CODIFICATI IN MODO DA INDICARE IL
TASTO DA PREMERE ASSIEME A 'SHIFT' O
'CBM'. ES. IL CUORICINO E' CODIFICATO
CON {SH S}.
 UN NUMERO DENTRO LE PARENTESI INDICA
LE VOLTE CHE IL TASTO VA PREMUTO.
```

**Listato 1. Programma BASIC per la misura della frequenza.**

```
50 IF PEEK(52000)<>120 THEN LOAD"MISROUT
   INE",8,1
60 POKE 45,100:POKE46,20:CLR
100 02=985248:REM CLOCK DI SISTEMA
110 TA=61578:REM 1/16 DEL CLOCK DI SISTE
    MA =$F08A
120 F$="{HOME}{CUR.GIU}{6 SPC}FREQUENZA:
    ":H$= HZ  ""
130 F1$="{HOME}{CUR.GIU}{9 SPC}**** PIEN
    O. ***{9 SPC}"
160 T1=138:T2=240:REM BASSO/ALTO TIMER A
170 T3=16:T4=0
180 CN=52000:FU=708:REM OBJ-ROUTINE
190 DIM TB(32),T1(16),T2(16),T3(16),T4(1
    6)
250 GOSUB 2000
300 GOSUB 1000
499 REM ***MISURARE***
500 POKE 704,T1:POKE 705,T2:POKE 706,T3:
    POKE 707,T4
510 SYS CN
520 IF PEEK(FU)<>255 THEN PRINT F1$:GOTO
     770
530 PU=65535-(PEEK(704)+256*PEEK(705))
540 FR=PU/TB
570 PRINT F$FR;H$;
770 IF PEEK(653) THEN 300
780 GOTO 500
```

*segue*

*segue*

```
999 STOP
1000 PRINT"{CLR}{2 CUR.GIU}"
1010 PRINT"{7 SPC}F R E Q U E N Z M E T
     R O"
1020 PRINT"{CUR.GIU} INTERVALLO CONTEGGI
     O:"
1100 PRINT" A) 1/128 SEC{3 SPC}MASSIMO 6
     5535 CICLI"
1110 PRINT" B) 1/64 SEC"
1120 PRINT" C) 1/32 SEC FORMERANNO L'INT
     ERVALLO DI{2 SPC}CONTEGGIO"
1130 PRINT" D) 1/16 SEC"
1140 PRINT" E) 1/8 SEC"
1150 PRINT" F) 1/4 SEC"
1160 PRINT" G) 1/2 SEC"
1180 PRINT" H) 1---SEC"
1190 PRINT" I) 2---SEC"
1200 PRINT" J) 4---SEC"
1210 PRINT" K) 8---SEC"
1220 PRINT" L) 16--SEC"
1230 PRINT" M) 32--SEC"
1240 PRINT" N) 64--SEC"
1250 PRINT" O) 128 SEC"
1300 GET Q$:IF Q$="" THEN 1300
1310 Q=ASC(Q$)-65:IF Q<0 OR Q>14 THEN 13
     00
1320 TB=TB(Q):T1=T1(Q):T2=T2(Q):T3=T3(Q)
     :T4=T4(Q)
1330 PRINT"{CUR.GIU}*****PER MODIFICARE
     PREMERE 'SHIFT'*****"
1399 RETURN
2000 FOR S=0 TO 14:READ TB(S),T1(S),T2(S
     ),T3(S),T4(S):NEXT
2599 RETURN
10080 DATA.0078125,17,30,0,0:REM PER .12
      5
10090 DATA.015625,17,060,0,0:REM PER .25
10100 DATA.03125,069,120,0,0:REM PER .5
10110 DATA.0625,138,240,0,0:REM PER 1
10120 DATA.125  ,138,240,1,0:REM PER 2
10130 DATA.25   ,138,240,3,0:REM PER 4
10140 DATA.5    ,138,240,7,0:REM PER 8
10150 DATA 1    ,138,240,15,0:REM PER 16
10160 DATA 2    ,138,240,31,0:REM PER 32
10170 DATA 4    ,138,240,63,0:REM PER 64
10180 DATA 8    ,138,240,127,0:REM PER 1
      28
10190 DATA 16   ,138,240,255,0:REM PER 2
      56
10200 DATA 32   ,138,240,255,1:REM PER 5
      12
10210 DATA 64   ,138,240,255,2:REM PER 1
      024
10220 DATA 128  ,138,240,255,4:REM PER 2
      048
```

**Listato 2. Programma in linguaggio macchina per la routine di misura**

```
Nome: MISROUTINE   CB20 CC0D

--------------------------------
CB20:78 A5 01 8D D4 02 A9 35  CB
CB28:85 01 AD FE FF 8D D0 02  2D
CB30:AD FF FF 8D D1 02 8D D1  95
CB38:02 AD FA FF 8D D2 02 AD  A2
CB40:FB FF 8D D3 02 A9 B9 8D  88
CB48:FE FF 8D FA FF A9 CB 8D  A0
CB50:FF FF 8D FB FF A9 90 8D  DC
CB58:0E DC A9 D8 8D 0F DC A9  72
CB60:B8 8D 0E DD AD C0 02 8D  22
CB68:04 DC AD C1 02 8D 05 DC  D8
CB70:AD C2 02 8D 06 DC AD C3  36
CB78:02 8D 07 DC A9 FF 8D C4  F9
CB80:02 8D 04 DD 8D 05 DD A9  D1
CB88:7F 8D 0D DC 8D 0D DD A9  B9
CB90:82 8D 0D DC A9 D9 8D 0F  75
CB98:DC A9 91 A2 B9 A0 A8 58  F5
CBA0:8D 0E DC 8E 0E DD AD 05  CE
CBA8:DD D0 FB AD 04 DD D0 F6  02
CBB0:78 A9 01 8D C4 02 4C CC  16
CBB8:CB 78 8C 0E DD AD 04 DD  BB
CBC0:8D C0 02 AD 05 DD 8D C1  DD
CBC8:02 68 68 68 A9 90 8D 0E  97
CBD0:DC A9 D8 8D 0F DC AD D0  99
CBD8:02 8D FE FF AD D1 02 8D  ED
CBE0:FF FF AD D2 02 8D FA FF  1D
CBE8:AD D3 02 8D FB FF A9 40  98
CBF0:8D 05 DC A9 7F 8D 0D DC  BE
CBF8:8D 0D DD A9 81 8D 0D DC  2B
CC00:A9 01 8D 0E DC AD D4 02  E1
CC08:85 01 58 60 53 00 FF 00  65
```